TEI：それはどこからきたのか，そして，なぜ，今もなおここにあるのか？

1. 背景と理論的根拠　Nancy M. Ide
2. どのようにして問題を解決しようとしてきたか　C. M. Sperberg-McQueen
3. なぜTEIは今もなおここにあるのか　Lou Burnard

※本稿は，Digital Humanities Conference 2017（於モントリオール）で行われたADHOアントニオ・ツァンポリ賞の記念講演の内容に手を加えたものである．[1]

1. 背景と理論的根拠

ナンシー・イデ

この講演で私たちは，「TEIはどこからきたのか，なぜ今もなおここにあるのか」という問いに触れたい．まず私は，このプロジェクトの背景と合理性を概観する．すなわち，どのようにしてTEIが始まったのか，なぜTEIが始められたのか，どのような問題の解決が試みられたのか，についてである．

まず，1980年代半ばのヒューマニティーズ・コンピューティングの状況を整理することから始めてみよう．当時のカンファレンスの予稿集を見ればわかるように，ヒューマニティーズ・コンピューティングにおける大多数の研究は，語彙のリスト，著者同定，文体研究のようなテーマ――おそらく当時より複雑な形ではあるにせよ，今日も続けられている身近な研究カテゴリーに向けられていた．計算機を扱う人文学者を手助けするためのソフトウェアはいくつか存在していた．多くの研究が用語索引に依存するために，かなりの用語索引ソフトウェアがあった．OCP（オックスフォード用語索引プログラム）は大型汎用計算機用のプログラムだったが，のちにパソコン用にMicroOCPとして再販された．WATCON（ Waterloo Concordance package）も汎用の用語一括索引プログラムのパッケージであった．ARRAS（Archival retrieval and analysis system）も同様に汎用計算機用のプログラムだったが，インタラクティブである点が最初の2件とは異なっていた．トロント大学で開発されたTACT（Text Analysis Computing Tools）プログラムはマイクロコンピュータ用のインタラクティブな用

[1] この日本語訳は，以下のような担当で行われた．
第一章　小風綾乃（お茶の水女子大学大学院人間文化創成科学研究科西洋史学博士課程）
第二章　小風尚樹（東京大学大学院人文社会系研究科西洋史学専門分野博士課程）
第三章　中村覚（東京大学情報基盤センターデータ科学研究部門）
付録　小林拓実（東京大学大学院人文社会系研究科欧米系文化研究専攻）
監訳　王一凡（東京大学大学院教育学研究科，永崎研宣（一般財団法人人文情報学研究所）

語索引パッケージで，ブリガムヤング大学で開発されたWord Cruncherパッケージも同様であった．

用語索引のパッケージによって，ユーザは，着目した語，単語のパターン，特定の語の組み合わせを明確にできるようになったかもしれない．ツールとしては，文字，語，フレーズの頻度一覧を作るものや，語のタイプやトークンに関して統計を算出するものがあった．後者には，テクスト全体に対するトークン比（Type/Token Ratio）やテクストの最初のn語に対するトークン比の累積計算のようなものがあった．検索語との結び付きが強い順に共起語を並べるツールもあれば，検索した語句のテクスト中での分布を視覚的に表示して，一目で主題や文体のパターンがわかるようにしたものもあった．概してテクストにおける語の使用を中心に，多くの，非常に多様な統計処理が行われていた．

これらのプログラムの制作者とユーザは，テクストに関する情報やその構造の情報を含んだまま，テクストを電子的な形式で表現する方法を求めていた．それは当然ながら，独自方式の乱立を招いた．TEIが設立される前の少なくとも20年，もしかすると30年もの間，この状況が続いていた．そのようなソフトウェアの場合，テクストを表現するには多くの問題を解決する必要があった．例えば特殊文字の表示（多くの用途において，アラビア数字・大文字のA-Z・多少の句読点以外のすべての文字が「特殊」とみなされていたように思う），テクストの論理的な区分の表現（章や段落，幕や場面・発話，詩の行など），そして，当然のことながら，研究者が自分の取り組んでいるテクストに付記したいと望むあらゆる分析や解釈の情報も．一部のテクストに関しては，何らかの形式でテクスト校訂情報を提供することが望ましかったが，その種の情報を線形［つまり，一続きのプレーンテクスト］に落とし込むことは難しく，したがって，テクスト校訂情報は，しばしば，単に省略されるだけになってしまった．他にも様々な問題があるが，とにかく，研究者がエンコードを必要とするかもしれない情報と，エンコードする必要を誰も感じないであろう情報とを分ける明確な境界は存在しないのである．

その様子がどんなものか教えてくれる事例をいくつかあげてみよう．

あるソフトウェアでは，引用のための参照情報がテクストの各行冒頭に置かれることを期待しているとしよう．そしてもちろん，電子テクストにおける改行は一次資料として用いた印刷版のテクストにおける改行と同じであることを期待しているとしよう．この場合，テクストの最初の行は以下のようになるだろう．

VirAen01001arma virumque cano, Troiae qui primus ab oris

これとは別の，COCOAと呼ばれるフォーマットの場合には，やや広く知られたものだったが（初期の用語索引プログラムに由来する名で，OCPでサポートされたために広く知られた），テクストの冒頭は以下のようになっていた．

```
〈W Shakespeare〉
〈T Merchant of Venice〉
```

```
〈A 2〉
〈S 6〉
〈C Graziano〉
This is the penthouse under which Lorenzo ...
```

ここからわかるように，COCOAは，メタデータを囲むためのXMLのような括弧と，何らかの意味を持つ情報を記述するための一文字のコードを使っている．後の，たとえばTACTのようなソフトウェアでは複数文字のコードを使えることが多かった．ここでは，〈W〉は著者（シェイクスピア）を指しており，〈T〉は著作のタイトル，〈A〉と〈S〉は幕（act）と場（scene），〈C〉は話し手（登場人物の名前），〈L〉（ここには登場していないが）は引用のための行番号である．

これらのテクストの表現形式は，概して，その時代のハードウェアとソフトウェアの制約に非常に大きな影響を受けていた．それらは往々にして互いに似ても似つかず，これらの形式の一つでエンコードされたテクストを他のものに変換することは大変な仕事であった．しかしあるソフトウェアを使ってエンコードしていたテクストを別のソフトウェアで使いたいのならば，このような変換は必須であった．かつて，完全な変換は不可能な場合もあった．COCOAの例にはもう一つの例では扱えない記録情報が含まれている．もしCOCOA形式からそちらに変換した場合には，その情報は失われるだろう．変換を自動化することもまた難しかった．電子テクストのユーザは，かなりの時間を，変換問題と悪戦苦闘したり，新しいソフトウェアで動くようにテクストを人力で編集したりするために費やしていた．

状況はほとんどいつも混乱状態であり，1987年のポキプシーでの会議では，その状況を文字通り「カオス」と呼んだ人がいた．

人々がその問題を問題だと認識していなかったわけではない．実際，IBMがかなり初期に開催した会議では，「文学データ処理 literary data processing」を取り上げた言語学者マーティン・ケイ，現在は計算言語学コミュニティにおける著名な人物だが，彼は「外部ソースから受け取るあらゆるテクストが従っていると見なしうる標準的なコード」があるべきだと主張した．[1]

彼の標準化の要求に応えるものは存在しなかった．しかしその考えは70年代から80年代初期を通じて主張され続けていた．1977年のサンディエゴの会議や，1980年にピサでアントニオ・ツァンポリが招集した会議を経て，人々は何らかの標準的なものを定めることをめぐって議論を続けた．

しかし解決すべき難問は，そのような標準のあるべき姿などではなく，そのような標準を作ることが本当に可能なのかということであった．多くの人々は不可能だと感じていた．なぜなら人々の同意を得られない可能性が高いと考えたからである．誰もが満足するような標準形式にたどり着く道などありえないのである．

それゆえ，1980年代後半，マーティン・ケイが問題を指摘してから20年を経ても，依然としてこのテーマに関する多くの議論が行われていた．そして，その時は来た．

1987年，「人文学におけるコンピュータ利用に関する国際会議」（当時は コンピュータと人文学学会（ACH, Association for Computers and the Humanities, 1978年米国にて設立）の年次大会であった）がサウス・カロライナ大学で開かれた時，不思議な巡り合わせがおきた．マイケル・スパーバーグ＝マックイーンと私は全米人文学基金の代表者であったヘレン・アグエラを交えて標準化の問題について議論した際，彼女は，このテーマを議論するために人々が一堂に会するワークショップを開くための臨時費用を申請するよう私たちに勧めてくれた．また，その論点のひとつが，電子テクストの標準的形式が実現可能かどうかだった．

ここからはご存じのとおりTEIの歴史ということになる．デイビッド・バーナードとルー・バーナードとともに，その夏，マイケルと私は臨時費用のための申請書を書き，採択された．会議はポキプシーにあるヴァッサー大学で，1987年11月12・13日の開催となり，当時存在した一握りのテクスト・アーカイブズ，たくさんのヒューマニティーズ・コンピューティング・センター，専門組織からの代表のほか，この問題に関する事柄に深く関わっていた人々も加え，世界中から32名が結集した．ACHはこのイベントの公式スポンサーであり，当然のことながら［欧州における同様の学会である］「文学研究と言語学におけるコンピュータ利用学会」(ALLC, Association for Literary and Linguistic Computing, 1973年英国にて設立，現在はEADH (European Association for Digital Humanities))から代表者を数人招待していた．しかしアントニオ・ツァンポリは例のごとくこの会議について友人ドン・ウォーカーに伝えた．ドンはコンピュータ言語学会の事務局長を長らく担当していた人であった．ドンは私を呼び，「私もそこに居なければならない」と言った．そして，会議室の許容人数が危惧されたものの，私たちは彼を招待者リストに加えたのである．

Attendees at the 1987 Poughkeepsie meeting
1987年ポキプシー会議の参加者

私たちの計画はちょっとした災難に見舞われた．アメリカ北東部では，11月11日に猛吹雪が発生しており，道路の移動が困難な状況だった．ポキプシーはニューヨークのやや北に位置しており，ニューヨークのJFK空港に到着してもタクシーに乗ればたどり着けるような場所ではなかった．参加者達，とくにニューヨークからポキプシーへの鉄道があることに気づかなかった人達は，ヴァッサーにたどり着くのに大変苦労した．ヨーロッパからの参加者たちはかなりの人数がJFK空港で立ち往生したが，またしても，アントニオ・ツァンポリがその問題を解決した．彼はワンボックス・カーの運転手を説得し，立ち往生していた全員を乗せてポキプシーへと連れてきたのだった．最終的に，全員が到着した ．図1は，ヴァッサー大学センターで撮影した集合写真である．講演者とアントニオ・ツァンポリ（私たちが受賞した賞は彼の名を冠して創設された）がわかるように示している．

ワークショップでは2日間の激しい―非常に激しい―議論が行われ，最終的に二つの点で全員の合意を得るに至った．一つは，テクスト・エンコーディングの一般的な手法の必要性であり，もう一つは，後に「テクストのエンコーディングと交換に向けたTEIガイドライン」となるものの開発を支える一連の基本的な原則であった．これらはポキプシー原則と呼ばれるものである．（頭韻を踏むことが好まれるので，ヴァッサー原則とはならず，ポキプシー原則（プリンシプル）となった．）

ポキプシー原則は後で詳述するが，ここでは要点をまとめてみよう．動機のいくつかはすでに説明した通りである．ハードウェア上の制約は当時あまりにも重要だった．既存のソフトウェアは，大抵の場合，プログラマでない人たちにとってはインストールして使うのは非常に難しいものであった．エンコーディング（内容）のメタデータと説明文書は皆無のことがほとんどで，もし誰かが説明文書かメタデータを提供しようとしても，多くの形式において，置き場所を見いだすのは困難であった．文書の見た目を志向する（テクストがどう見えるかを記述する）手続き的マークアップを，記述的マークアップ（テクストが何であるかを記述する）から区別するという考え方は，GML（Generalized Markup Language），LaTeX，そして1986年におけるSGML（Standard Generalized Markup Language）などを通して，何年もかけて徐々に確立されていった．1987年11月，ポキプシー会議と同じ月に，ジェームズ・クームズら数名が，そのテーマに関する重要な論文をCommunications of the Association for Computing Machinery上で発表した[2]．ポキプシー会議の議論の文脈もまた，単一の標準にたどり着くことなど不可能だろうという当時の見解に強く影響されていた．

最終的な文書では明示的に述べているわけではないが，私たちの仕事にとって不可欠であったのは，クームズらが出たばかりの論文で推奨した考えだった．つまり，私たちは，記述的であり，かつ，命令的ではないマークアップを提供しようとしたのである．そして，ソフトウェア的・ハードウェア的な要件をきっぱりと無視して文書中にメタデータを入れる場所を用意することと，学術の営みにおいて必須の，あるいは望まれる情報の表現に的を絞ることも決定した．（1987年当時は，それは想定するより難しいものだった，何故なら今日ほどマシンの性能がよくなかったからである．）

おそらく自明だが言及に値することとして，このガイドラインは，何らかの用途に特化されたソフトウェアがなくとも使えるように，シンプル，明快，具体的であることをもう一つの目標としている．これは私たちの活動の信念であった．というのは，そこには他のソフトウェアに依存しないデータ形式の記述と，その形式をサポートするソフトウェアを実現しようとする希望とが含まれていたからである．同時に，私たちは効率的なテクスト処理のための厳密な定義を可能にしたかった．これは明らかに，単純さと厳密さや，効率と利便性あるいはソフトウェア非依存性を天秤にかけねばならず，そこに線引きをすることは非常に難しかった．

ポキプシー原則は，この会議でテクスト・アーカイブズの代表者たちから挙げられた懸念事項を随所に反映している．彼らは，運用システムとエンコーディング方式の開発と，そのスキームに従ってエンコードされた文書を処理することができるソフトウェアの作成に多大な努力と時間を費やした．そして，新たな標準的な表現形式がすぐに手元にある既存のものを遡ってすべて変換したいという要求を引き起こす，ということと，そのような遡及的な変換には大きなコストがかかるだろうということを恐れた．会議では，彼らの懸念はとても大きな影響をもたらすこととなり，そして，合意に至るために，ポキプシー会議の主要な結論は次のようなものとなった．すなわち，このガイドラインはエンコーディングの原則を「提案する」ことを目的とし，厳密な規範的標準の設定は棚上げすることとした．それらは「標準」ではなく，単なる「ガイドライン」ということになった．現在，その原則はあまりにしばしば記されているため，それが自然な状態であるかのようにみえるかもしれないが，実際のところ，ポキプシー会議において意識的になされた決定なのである．

ポキプシー原則はこのガイドラインをデータの「交換」のための標準的な形式を提供するものとして記述している．その言葉の選択が暗黙的に示すのは，あるリポジトリの中では既存のテクスト・アーカイブズが安心して既存のテクスト・エンコーディング方式を利用し続けられるということであり，すなわち，ローカルなものへの準拠は必要ないということになる．言い換えれば，TEIはピボット形式，あるいは媒介言語の役割を果たすものである．もし自分のテクスト・リポジトリが異なるスキームを採用しているのなら，内部スキーマからTEIに文書を変換し，TEI文書を内部スキーマに変換するためのソフトウェアを作ることになるだろう．しかし，異なるソフトウェアを利用したり他の組織とテクストを交換したりするために，何十にものぼる方式への変換ソフトウェアを作る必要は「ない」ということになる．

この原則では，ガイドラインは目指す形式のための推奨文法を定義す「べき」だと述べている．しかし注意深く読むと，正確な文法がどうあるべきかということについての最終的な決断を提示しているわけではないことに気づくだろう．多くの人々はわずか1年前の1986年にISO標準になったSGMLを推奨していたが，当時SGMLは皆に受け入れられていたわけではなかった．最終的に，それが自分たちの基準にかなうという総意が得られてTEIがSGMLの利用を義務化するのは，しばらく経ってからだった．

この原則が「推奨する文法」に言及する時，特筆すべきことは，実際のところ，厳密な意味での文法，すなわち，一連のデータの物理的な形式としての文法と，セマンティクス，すなわち，あるエンコーディング方式におけるラベルやエレメント，あるいは，何であれ意味を持たせたい単位の意味としてのセマンティクスとの両方を意味しているということである．

（たとえ言葉が明瞭でなかったとしても，私たちがこのことを意識していたことは明らかだった．）周知のように，SGMLは，様々な種類のタグと区切り文字による一つの文法的な形式のみを規定しており，XMLも同様である．SGMLとXMLのユーザは文書型宣言あるいはDTD（もしくは，後には別の記法によるスキーマ）の形式で文脈自由文法を定義することができる．DTDは，どのエレメントがどこに現れることができるのか，そしてそれが何回現れることができるのか，等を指定するものである．しかし，TEIガイドラインの90%，もしくは95%はそのスキームのセマンティクスに当てられている．それは，ラベルやエレメント名の定義，何らかの形式仕様記述言語ではなく，注意深く作り上げられた自然言語の文章である．

もう一つの原則は，アーキビストの懸念を反映して，ガイドラインには人々が守るべき最低限の規則は含まれるが，「追加」しなければならないものは一切ない，としている．新たにエンコードされるテクスト――それは必ずしもアーカイブズにすでにあったレガシーなテクストとは限らないが――は，エンコーディングそのものに関する記述的かつ書誌的な情報やメタデータを含むべきである．これらは当時の一般的な慣例を凌駕するものだったといっても過言ではない．当時はだれもそのような種類の背景情報，とりわけエンコーディングそのものについてのメタデータを提供してはいなかった．

さらに，スキームを拡張する方法があるべきとされた．

これらのすべては，現在では当たり前のように聞こえ，だれもが慣れ親しんでいることだが，1987年の時点では自明ではなかったのである．

もうひとつの積極的な関心事は，理論の多重性である．それは，ポキプシー会議の何年も前から人々の口に上っていたものである．同じ事象についてすら，人々の間で，何をどのようにエンコードすべきか異論が出ないことはほとんどなかった．私たちは研究者の知的な自律性を失わないようにすることを望んだが，しかし同時に表面的な事柄における無意味な差異を回避するために十分な手引きを提供することをも望んだ．そこではまたひとつ，柔軟性と規範性のバランスを取る作業が必要になったのである．

これらの要件に対するTEIの解決策は，DTDとして表現された―ゆえにエンコーディングのための具体的なルール集となる―文書の固有の文法である．しかし，必要性が感じられるならば，同じものをエンコードするための代替手段をも提供している．

ポキプシー原則のひとつは実現されることはなかった．私たちは当初，エンコーディング方式，とくにそのセマンティクスの記述のためのメタ言語を定義することを意図しており，そして，このメタ言語で新しいスキームと広く多様な既存のエンコーティング方式の両方を記述し，それによっていくつかの変換を可能にすることを望んでいた．これは実現されることはなかった．理由のひとつは，TEIが発展するにつれ，テクスト・アーキビストたちがSGMLと新たな形式に抱いていた不安がおさまっていったからだ．SGMLは，広く受け入れられた．もうひとつの理由は，データの爆発的増加であった．当時実際に存在していたアーカイブズは，大規模で重要なものではあったものの，1990年代初期になって出てきたものに比較すると，そのテクストの量はわずかなものであった．その後のことは言うまでもない．

テクスト・エンコーディングの形式を記述するためのそのような普遍的な表記が開発されなかった第三のやむにやまれぬ理由は，そのようなものを開発する方法を当時誰も知らず，そして現在も誰も知らないためである．したがって，それが決して実現しなかった理由の一つは，こう言っていいならば，まだ誰も作ってないからである．そのようなスキームを創り出した人はかつて誰もいなかった．もしあなたが出来るのなら，話を聞きたい人はたくさんいるだろう．

ポキプシー会議の後，プロジェクトの運営は，3つのスポンサー組織それぞれからの代表者2名ずつを含む運営委員会に委任された．3つのスポンサー組織とは，ACH，ALLC，ACL（Association for Computational Linguistics）であった．その後の数年を通じて，このグループはTEIの作業を支援するために，ヨーロッパと北アメリカの両方で100万ドルを超える資金を集めた．この運営委員会は，プロジェクトを自立可能にするためのTEI協会が2000年に設立され，3つのスポンサー組織がガイドラインの責任をこの新しい協会に委譲するまで，このプロジェクトを監督した．

標準的なエンコーディング形式の必要性についての議論と，それを作成しようとする多くの先行する試みには長い歴史があった．それまでの取り組みでは成功しなかったものが，なぜTEIでは成功したのだろうか．

私が思うに，そこにはいくつかの理由が存在する．私たちはたいてい，あるものを他のものに変換しようとするときに頭を悩ませた経験から，エンコーディングの問題点についてより多くのことを知っていた．私たちは，何が必要なのかをより深く理解することから始めた．そして，それまでに開催されたどの会議よりも，はるかに堅実な組織，研究者，研究所の代表をポキプシーに集めた．SGMLはちょうどいいときにちょうどいいツールを提供してくれたようだ．前にも述べたが，巡り合わせがよかったのである．

なぜ当時TEIが成功したのかというもう一つの理由は，とくにこのツァンポリ賞の文脈で言及されるのがふさわしい．なぜならアントニオ・ツァンポリがTEIを支持し，支援をしてくれたがために，TEIは成功したという側面がある．アントニオが何かを起こしたいと望んだとき，彼は無視できない影響力を持つ人物だった．

最後に，TEIに関するいくつかの個人的な感想を述べておきたい．

何年もの間エンコーディングとデータの表現の課題に取り組むなかで，TEIがいかにして多くの根本的な課題に応えていたかを知ることは興味深く，素晴らしくさえあると気がついた．

- ピボット形式というアイデア
- 異なる手法によるテクストのエンコーディングを許容することによって理論の多重性の問題に対処しようとすると同時に，できるだけの箇所で一貫性を確保しようとするTEIの試み
- SGMLのような既存の標準の利用
- 文書自体に書誌情報とエンコーディング方式の記述を含むこと

さまざまな表現形式を扱う中で，これらの原則の多くは今日でも生きている．

第二に，TEIは当時，そしておそらく今もなお，エンコーディング形式のためのセマンティクスを提供しようとするかつてない広汎な試みであるように思う．今日に至るまで，セマンティクスを定義することと，そのセマンティクスをスキームを超えて継続的に適用することは未だに相互運用性に対する大きな障害となっている．非常にたくさんのオントロジーや分類法が存在しており，誰もが自分たちの名付け方ととらえ方を持っているため，共通の参照点がなければ相互運用性を実現するのは非常に難しい．

最後になるが，TEIの規模，対象範囲，そして影響がポキプシー会議で私たちが想定していたものをはるかに超えてきているように思う．30年の時を経て，私たちが少し歳を取って，願わくば少し賢明になって，今ここにいられること，それを祝福できること，そしてTEIコミュニティ全体を代表して，TEIの功績のためにアントニオ・ツァンポリ賞を受賞できたことを大変うれしく思う．

2. どのようにして問題を解決しようとしてきたか

C. マイケル・スパーバーグ＝マックイーン

ナンシー・イデの発表は，TEIがどのような背景から生まれたのかについて述べ，TEIが解決しようとした一連の問題について概観したものであった．この発表での私の目標は，これらの問題を私たちがどのように解決しようとしたのかについて説明することである．このために昔を振り返り，今の私にとってTEIの本質を設計するための決断にはどのようなものがあったのかを確かめようとした．それらを要約して以下に列挙してみよう．

- 記述的なマークアップを用いる
- ドキュメントの中でメタデータを提供する
- 過剰生成する
- 拡張可能性を設計する
- モジュール化する
- オープンな標準［規格］を用いる
- 文書の構造を利用する
- プログラムではなくデータフォーマットを作る

記述的なマークアップを用いる

第一に，そしておそらく最も重要だったのは，記述的なマークアップを用いるという決定だった．ほかの様々な要素の中でも，記述的なマークアップというのは次のような信念を必然的に伴うものであった．すなわち，文書というのは重要な内部構造を有しており，わかりやすい形で処理ソフトウェアに対してその構造を示すことには価値がある，というものである．記述的なマークアップは，「ソフトウェアがドキュメントのこの一部をどのように処理すべきか」という質問に直接答えようとするものではなく，その代わりに「ドキュメントの

この部分は一体何なのか」という質問に答えるものである．記述的なマークアップは，宣言的なものであり，命令的なものではない．これは，ソフトウェアに指示をするというよりは，むしろテクストの説明をするものなのである．

加えて，当時の定義にしたがえば，記述的なマークアップは，ドキュメントの文法という概念を提供し，私たちが自分たちのデータに制約をかけたり検証したりできるようにするものであった．データの破損・回線ノイズ・誤字脱字が些細な問題として片づけられないということを痛感させられた私たちは，完全な自動処理でそれらのエラーをいくらかの割合だけでも検出してくれることがきわめて有用であることを理解していた．

一つの文書の文法を解析すると，一つのツリー構造が得られる．もしその文書の文法がポインタをも提供する場合には，スパニングツリーを構成する有向グラフが得られることになる．

つまり，具体的に言えば，SGMLを採用することにしたのである．

ナンシーが述べたように，ポキプシー会議の参加者にはSGMLに疑念を表明する人もいた．しかし，SGMLを研究した結果，私たちはSGMLがむしろこのこのエンコーディング方式の要件を満たし得るものであり，そしてSGML準拠の文法は私たちが場当たり的に開発するよりも良いものになりそうだという結論に至った．そこで，私たちはSGMLを方式の基礎とした．SGMLは国際標準であるという点に優位性があり，SGMLが定義するデータフォーマットは特定のプログラムに束縛されるものではない．ソフトウェアに依存しないというこの特徴は，アプリケーションや個々のベンダーからの独立性をより容易に実現できるのであり，それによって特定のベンダーに囲い込まれないようにすることができる．また，ソフトウェア・ベンダーではなくユーザがデータを保有するということへの技術的な支柱ともなる．SGMLの機能は，マークアップ方式の文法を定義することにとどまる．すなわち，SGMLの文脈では，SGMLエレメントとその意味の具体的な集合（セット）を定義する責任はSGMLのユーザが持つことになる．

ユーザが定義する特定のエレメントセットは，（どこまでを文法と（いう概念で）呼ぶか）当然のことながら文法にも影響を与える．しかし，一番簡単な分け方をすれば，SGMLがエンコーディング方式の文法を提供し，ユーザがそのセマンティクスを提供することになる．

古今のほとんどの文書形式には，少なくともこうした性質のいくつかが欠けており，LaTeXのようなほかの記述的なマークアップでさえ，自動検証の機能が欠けている．

したがってSGMLは，データの長期保存，ソフトウェアへの非依存，そしてデータの正確性を重視する人にとっては，競争相手になりうるものがまったくなかったのである．私の見解では，現在でもその状況は依然として変わっていない．

文書の中でメタデータを提供する

TEIの設計にあたって，二つ目に重要な決断は，<teiHeader>エレメントを必須とすることにより，文書の中でメタデータを提供することであった．これはとても単純な動機に基づい

ている．すなわち，メタデータを別ファイルにしておくと，失われてしまう．外部のメタデータは常に失われてしまうのである（もちろん，ほとんど常に，ということだが，事実上，常に失われてしまう）．私たちがこのことを知っていたのは，当時ルーが10年以上にわたってOxford Text Archiveの運営をし続けていて，アーカイブに寄贈される資料のどれほど多くのものがその文書についての説明のないままに提供されるかを知っていたからであった．そして，私たちは二人ともコンピュータ・センターのヘルプデスクに座り，［データの入った］テープを持って質問しに来る人に対応していた．彼らは，そのテープを読んだり理解したりすることを助けてほしいと尋ねてくるのだが，その際にその文書について説明するものを持っていないのである．あるいは，そういった説明の存在や必要性といった発想すら持っていないこともしばしばであった．

そこで私たちは，この社会科学データアーカイブを教訓とすることにした．彼らが定義していたデータフォーマットの中では，すべてのデータセット（つまりすべての体系的なデータ・コレクション）は二つの要素に分かれていた．データそれ自身が，そのデータセットの一つの要素であり，データの構造や内容の説明としてのコードブックがもう一つの要素である．これらのフォーマットにおいては，もしコードブックが存在しなければ，データセットも存在しないも同然である．ヨーロッパにおける出版社は，同じような教訓を1450〜1500年の間に学んでいた．すなわち1500年以前には，非常に多くの書籍がタイトルページや，何らの前付けにあたる部分も伴わずに出版されていたが，1500年以降には，タイトルページやその他の前付けのような形式で書籍内にメタデータが付されて出版されることがヨーロッパにおける標準となった．資料にメタデータを組み込んだ形で提供することが，TEIの設計における重要な要素の一つなのである．

過剰生成する

第三に，私たちのスキーマ向けのドキュメント文法を定義するにあたって，私たちはある選択を意識的に行った．よくない文法には二つの方向性がある．一つは，あまりに厳密になりすぎた結果，本来含めるべきものを含み損ねてしまうことである．（言い換えれば，実際の言語には存在する文章を生成し損ねることがあり得る）．あるいは，あまりに寛容にしすぎた結果，その言語には存在しない（もしくは存在すべきでない）文章を受け入れたり生成したりしてしまうことになる．いずれの誤りも文法を脆弱なものにしてしまうが，方向が違うのである．

もし過剰生成する文法の場合，文法に対する妥当性の検証があまり便利ではなくなってしまう．ある程度のエラーは正しいエンコーディングとして許容されることになるからだ．もしその逆だった場合，一般的でないテクストを研究する研究者は困ったことになるだろう．なぜなら，ガイドラインに従って自分たちが扱うテクストを利用するために，ガイドラインで定義されたスキーマや文書の文法を変更しなければならなくなってしまうからである．妥当性の検証と品質管理を気にする人々は，文書の文法についていくらかの理解があり，スキーマを変更するのに必要な技術を持っている可能性が高いように，私たちには思われた．それは不運にも珍しい構造を持つ不可解なテクストを研究することになった人々には，必ずしもあてはまらないだろうと考えた．そのため，私たちは，なるべく負担が分相応になるよう努めた．

拡張可能性を設計する

第四に，当初から私たちは，変更という考え方をTEIに盛り込むことにした．

SGMLとXMLの基本的な前提は，単一のマークアップ言語がすべての用途，すべてのユーザ，そしてすべてのアプリケーションを満足させることはないというものである．それが理由で，SGMLの開発者たちは，マークアップ言語というよりもメタ言語を開発したのであった．彼らは，自分たちがすべてのユーザを満足させるマークアップ言語を定義する立場にあるとは信じていなかったので，その代わりにユーザが自分自身のためのマークアップ言語を定義することができるようなメタ言語を開発したのである．SGMLのユーザは自分自身でルールを定義することになった．いまや，SGMLの文脈においては，TEIとはSGMLの（そしてしばらく後には，今のXMLの）ユーザの一つであり，TEIはまさに多くの特化されたエレメントの種類によって専門的なマークアップ言語を定義している．しかし，SGMLやXMLと同様に，TEIは非常に多様なコミュニティや，まったく予想だにしなかった一連の活用例やアプリケーションをサポートしようと努めている．

つまり，TEIにいる私たちは，当初からすべての人々，あらゆる種類のテクスト，それもすべての言語，あらゆる時代，そしてすべてのジャンルのテクストに対して有効な単一のエンコーディング方式を提供するというゴールを設定したと同時に，そのゴールを実現するのが不可能であることを当初から受け入れていた．この目標の実現不可能性を克服する方法とは，変更と拡張を許容することである．そうすることで，私たちの開発したスキームが何らかのテクストに対して不適切であることが明らかになった時に（この事態が必ず起こることを私たちは知っていた），そのテクストを扱うためにスキームを拡張することができるのである．

拡張と修正の可能性というのは，重要な政治的側面をも有している．もしあるユーザがTEIを修正できるとしたら，そのユーザは自分自身が持っていたテクストに適応するようにTEIを作り変えることができる．だから修正を許可するということは，専門的なデータ・コレクションの構築に長年を費やしてきた既存のテクストアーカイブが抱く懸念に対処するための手立てでもあった．

このような検討の結果，やや単純化して言えば，ガイドラインの最初の完成版として1994年に公開されたTEI P3文書において，TEIへの準拠がほとんど何も保証しないような形で準拠に関するルールが設計されたのである．TEI P3のルールの下では，スキームにおけるすべてのエレメントは再定義が可能である．その際，いくつかの制限は存在する（残念ながら，TEI P3の文章ではうまく説明できているとは言いがたいが）．それは，最終的なエンコード結果がなんらかのヘッダを持たなければならず（したがって，teiHeaderエレメントというのは再定義したり名前の変更ができるが，完全に取り去ることはできない），そして，そのヘッダには，例えば，一つのtitleエレメントにおいて，文書のタイトルを記さなければならない．titleエレメントは，なんら有用な内容を持たなければならないということはない．なぜなら，ナンシーが言及したように，テクストの遡及的なアップグレードを要求することはできないからである．もし，手元にあるタイトル不明のテクストをTEIに変換できるようにしたいと思ったならば，そうできるべきである．しかし，TEIのtitleエレメントは必須で

あるため，もしその作品のタイトルがわからないとしたら，エレメントの中身を空にするか，「タイトル不明」などと明記しなければならない．言い換えれば，タイトルを知らないことを認めざるを得なくして，少しの申し訳なさと，今後へのやる気を抱かせているのだ．

またTEI P3に準拠する際のルールにおいてユーザが行うすべての変更は，TEI P3それ自体の参照文書に用いられるスキーマで緩く定義されているタグ・セット・ドキュメンテーションという文書に記載されなければならない．もしこれらのルールに従うならば，TEIに準拠するあらゆる文書を利用する誰もが，エンコーディング，そして特に文書内に出現するであろうすべてのSGMLエレメントに関して，あらゆる詳細が記された説明を手に入れることになる．このエンコーディングの場合には，TEIガイドラインに記述されている無印のTEIを用いるか，変更者が説明を提供するカスタマイズしたエレメントを用いるかにかかわらない．

言い換えるなら，制限なく修正ができるように許可するのと同時に説明を用意するように求めることによって，私たちは，ナンシーが理論の多重性に関する議論で述べていたような，ユーザの知的な自由さと自己決定を最大化しようとしたのである．またそれと同時に，ドキュメントの再利用可能性，あるいは少なくとも未来のユーザに対するわかりやすさを最大化しようとした．私たちは，これがプラグ・アンド・プレイ方式（※接続したらすぐに作動すること）の相互運用性をより困難に，あるいは不可能にし得ることを十分意識的に理解していたにもかかわらず，TEIユーザによる自己決定を保証しようとする道に挑戦することを選択した．この決断は，プラグ・アンド・プレイ方式の相互運用性を望む人々にとっては不満のたまるものだと私は承知していたが，しかし，意識的に選択したのである．

元のテクストをありのままに伝えることが，あらゆるエンコードされたテクストの責務である．そしてそれを実現することは，エンコードする者の責任である．したがって，テクストのありのままの姿を表現するために必要とあらばTEIの方式に変更を加えることは，テクストをエンコードする者すべての権利であり責務なのである．

TEIは，時として秘密政府であると揶揄されることがある．もしそうなら，それは，全力で市民に自治を求め，そのために必要な手立てを保証するために，決して楽ではない道のりを歩んできた秘密政府なのである．

モジュール化する

TEIの設計における第五の主要な決定は，特別難しかったり議論を呼んだりしたわけではない．私たちは，TEIのガイドラインをコアとなるモジュールといくつかのモジュールに整理し，それらが特定の規則に応じて互いに結び付けられるようにした．ガイドラインには，共通のコアとなるタグ，いくつかの基本的なテクストの型，そしていかようにも組み合わせられる選択的なモジュール群がある．私たちはシカゴ風のピザというメタファーを用いた．つまり，ピザ生地の種類（基本的なテクストの型）を選び，好みのトッピング（任意のモジュール）をリストから自由に選ぶことができ，そして特段拒否しようとしない限りは，トマトソース（コアとなるタグとTEIヘッダ）が付いてくるという具合である．

オープンな標準を利用する

TEI設計に関して言及されるべき第六の決断は，SGMLのようなオープンな標準を可能な限り用いたということである．しかし，決して既存の標準に限定したわけではなかったため，必要な場合には新しい標準を設計することを厭わなかった．当初はSGMLを，そして後にはXMLを用いた．ほかにもユニコードや，URI（Uniform Resource Identifier）を活用した．一方で私たちは多くのものを発明した．そのうちのいくつかは，その後，より広汎な標準の発展を助け，それが当初のTEI専用の機能の代わりにTEIに採用されたために取り除かれた．いくつかの例については，言及する価値があるだろう．

- 1990年のTEI P1では，ハイパーリンクのための拡張ポインタの文法を開発した．これは後に，W3CのXPointer Specificationの取り組みに影響を与え，TEI P5では，拡張ポインタ記述は，XPointerのTEIプロファイルに取って代わられた．

- ほかの領域では，TEI P1は書字体系宣言（Writing System Declarations）のための特殊なドキュメントタイプも定義した．これにより，レガシーな文字のエンコーディングの説明の記述やアドホックな翻字方式が可能になった．いまや書字体系宣言は，大部分が，ユニコードを用いるというシンプルなXMLの規則に取って代わられている．これはナンシーが言及した現象の好例であり，1987年の段階では喫緊の課題であったことが，時代を経るにつれて差し迫ったものに思われなくなったということである．現在，当時ほどには重要でなくなったように感じられるのは，プロジェクト固有のアドホックな翻字方式をTEIが許容することである．というのも，今日，ほとんどのユーザは，ユニコードの良いサポートにより，さまざまな表音文字による代替翻字方式について理解したり見ようとしたりすることを望まないようになっている．技術的なレベルでは，ユニコードは皆の生活をずいぶんと簡単にしてきた．ユニコードが対応しきれない文字（これは避けられない．というのも，これまで人間が使ってきたあらゆるシンボルを搭載した完璧な一覧表の作成は，あらゆる学術的用途のためにエンコードされたテクストすべてにうまく機能するテクスト・エンコーディング方式の作成と同じぐらい困難かつ不可能だからである）については，現在のTEIでは任意の書記体のためにgエレメントを定義しており，これによってユニコードに登録されていない文字のエンコーディングができるようになっている．

- 私たちは，フルSGMLの場合よりもシンプルにTEI文書を解析できる，事前の相互確認なしに交換可能なTEIのフォーマットも開発した．この機能はXMLによって必要ないものとなったが，XMLはそのTEIのフォーマットに影響を受けたものなのである．

- TEI P3で定義されたもう一つの特殊用途の文書タイプは，SGMLの語彙のための参照ドキュメントであった．

- その後，これは特殊用途の語彙としての役目を終え，関連エレメントはメインのTEI語彙に統合された．

オープンな標準を用いるのと同様に，私たちは比較的オープンな方法で仕事をしていた．草案を無料で一般公開し，公開のメーリングリストを作り，そしてTEIの進展や計画に関す

る定期報告を，スポンサー組織が開催した学会などの場で公表した．その後のインターネットの発展や，分散型の共同作業のためのツールによって，その数年の間によりオープンな仕事のやり方を構築することが可能となった．このことが意味するのは，振り返ってみて，TEIの共同作業が普通の仕事と比べてどの程度異なっていたのかということを評価することが難しいかもしれない，ということである．

ドキュメントの構造を利用する

特定のタグセット設計におけるひとつの具体的な選択については言及する価値があるかもしれない．というのも，私たちは（スキーマをシンプルなものにとどめておこうと試みる中で），ドキュメントの構造を分析して有効活用する処理ソフトウェアの力に頼ることによって，区別すべきエレメント・タイプの数を減らすことにしたのである．SGMLそして後にはXMLのタグセットの多くには，構造における様々なレベルに対して，明確に区別できるエレメント・タイプがある．例えば，章・節・項・目，あるいはそれらに類するものである．そして，それらの構造的なユニットの中では，節の見出しを示すタグもまた同様にはっきり区別できる．つまり，章の見出し（chapter-head），節の見出し（section-head），項の見出し（subsec-head），目の見出し（subsub-head），などである．しかしながら，それぞれのレベルの見出しに対して，個々に名前を付けるというのは，ある意味で冗長である．例えば，章の見出しは常に章の見出しであって，目の見出しではない．それでもやはり他のタグセットは，containerエレメントを提供することなく，見出しのレベルを明確に区別している（それは，divやsectionエレメントが導入される前のバージョンのHTMLのh1，h2，h3エレメントの内容と同じようなことである）．

TEIのタグセットは，異なるレベルのセクションについて，div1， div2， div3など一般的に名付けたものを提供しているが，それぞれのレベルにおいて見出しはheadとタグ付けされる．headエレメントを処理する際に，第一レベルの見出しなのか，第二なのか，それとも第三レベルの見出しとして処理するかどうかを知るために，処理ソフトウェアはエレメントの文脈を分析しなければならない．例えばもし親要素がdiv1であれば，見出しは第一レベルの見出しであり，親要素がdiv2であれば，見出しは第二レベルの見出しである，という具合である．当時使用できた文書処理ソフトウェアのすべてが，そのような類の文脈情報を利用できたわけではなかったが，SGMLで利用できると確信していた．XSLTの発展が，さまざまな点において私たちの予測を実現した（そして，しのいだのである）．したがって今では，文脈に依存する必要があるかもしれないと予測しながら，TEIドキュメントを処理することは簡単である．なぜなら，XSLTやほかのXML志向の言語が，TEIが設計上求める文脈情報の類に簡単にアクセスできるようにしてくれているからである．

プログラムではなくデータフォーマットを作る

設計に関する最後の決断は，非常に広く影響力をもたらす結果となった．その決断は，一つのソフトウェアを作るのではなく，データフォーマットを定義しようと決めたことである．もしTEIがあるソフトウェアを規定して推進していたとしたら，データフォーマットこそがプロジェクトの主要な成果物であって，ソフトウェアが二次的な成果物，すなわちTEIのデ

ータフォーマットを使用して実行できるもののサンプルにすぎないということを慎重に警告しても，次のようなことになっただろうと私は今日もなお確信している．すなわち，TEIの外にいる大半の人々は「ああ，TEIね，あれはソフトウェアの類でしょう」と言っていただろうということ，そしてたとえそのような人の中に「あとは，ソフトウェアが使用しているデータフォーマットもTEIですね」と付け加える人がいようとも，対象となる聴衆の心中では，依然としてソフトウェアこそが最重要の成果物であり，データフォーマットは二次的なものであり，そしてTEIのデータフォーマットが普遍的で，ソフトウェアに依存しないフォーマットなのだとは受け入れられないだろうということである．データは，ソフトウェアよりも寿命がとても長い．人文学者はいつでも，何百年，あるいは何千年も経ったデータを扱っている．それにひきかえ，何千年も経ったソフトウェアを扱っている人などというのは，ほとんどいないと言っても過言ではない．

TEIが，P1からP4にかけての取り組みの一部としてソフトウェアを開発することを拒んでいたというのは，当然のことながら，既製のソフトウェアを望む人々にとっては非常に不満なことであった．あらゆる代替選択肢や特殊事例に対処するためにソフトウェアを開発するという課題を，私たちが無邪気に無視していたために，TEIはあるべき理想の形よりも大きく，より複雑に成長してしまった，と主張することは可能である[3]．しかしながら，便利なソフトウェアをまったく持たないことのリスクと効率的でソフトウェアに依存しないフォーマットを作らないリスクとの間には，相反する矛盾がある．私たちは，ソフトウェアよりもデータを選んだのである．

振り返ってみて，かつての決断は，いまも有効だろうか．

思い返してみるに，もし同じ状況で，もう一度これらの選択を行うことになったとしたら，私は同じやり方ですべての選択をしたいと思うだろう．今では，TEIを見るすべての人たちにとって，これらの選択のすべてが理想的な決断であるとは映らないかもしれない．しかし，私が思うに，今これらのすべてが良く見えるわけではない理由のひとつは，私たちが1987年の頃とは異なる状況に置かれていることである．そしてまたひとつの理由は，一般に，たとえ部分的な成功であっても，状況を変え得るということ，そしてその結果としてトレードオフにおける異なる選択肢の相対的なコストや利点についても変えてしまう可能性があるということである．ある意味では，TEIが十分に状況を変えたことによって，私たちがTEIの設計当初抱いていた懸念が，最早かつてほどは差し迫ったものに見えなくなったことから，TEIの成功の大きさを測ることができるとも言える．

しかし，もし1987年から状況が劇的に変わってきているとしたら，なぜ当時に始まった解決策が今も現役で，そして今日的にも有効なのだろうか．なぜTEIが，30年経った今でも，かくも異なった世界で，現在進行形でまだ存在しているのだろうか．これこそが，ルー・バーナードが自身の言葉で私の発表に対して答えようとする問いである．

3. なぜTEIは今もなおここにあるのか？

ルー・バーナード

ナンシーから聴いたように，TEIは非常に長い歴史を持っている． TEI P3が公開された後，私たちがガイドラインの普及に注力した時期があり，この時から，そのガイドラインは広く受け入れられるようになった．それがデジタル図書館の黎明期に伝染病のように広がった興味深い時期があった．

1990年代後半に，助成金だけに漫然と頼ることが不可能であることを認識した結果，助成金とは独立してTEIを進めるためのビジネスモデルを持つコンソーシアムを設立することが決まった． そして，準備期間を経て，現在のTEIコンソーシアムが2000年12月30日に設立された．それに続く数年間，私たちは最初にTEI P3をSGMLからXMLに移行することに注力し，ほぼすべての用例を更新するとともに 文書型宣言（DTD）を（当初想定していたよりもはるかに大きな一部を）書き換えることになった．

その後，選挙で選ばれた技術委員会は徹底的な改訂作業に着手し，2007年にはTEI P5として公開され，それ以来定期的に更新され続けている．

TEIが実際に誕生したのはかなり以前のことである． World Wide Web，DVD，携帯電話，ケーブルテレビ，Microsoft Word，イギリス海峡のトンネルよりも古くから存在する． テクノロジ（特にソフトウェア）が数年以上存続することは本当にきわめてまれなことである．それでは，なぜ，どのようにしてTEIが生き延びたのだろうか？これから，私はこの質問に答えることを試みたい．もしくは，少なくとも私が答えだと考えることについて，示唆的な意見を提供したい．

第一の理由は「テクストは私たち」だからである．

デジタル・ヒューマニティーズ，ヒューマニティーズ・コンピューティング，文学，あるいは言語コンピューティングについてのあなたの定義が何であれ，もしくは私たちのすることが何であれ，テクストは，あなたが行うことの核心であるか，あるいはあなたが行うことに対して少なくとも非常に意義のあるものだ． テクストは大切なのである．

私が意味する”テクスト”は，言葉通りの意味でのテクストである． つまり，頁画像，翻刻された文書，あるいは，これら2つの組み合わせのみについて語っているのでもなければ，注釈や解釈が付された翻刻文書やメタデータについてだけ語っているわけでもない．それらのすべてを意図している． それがテクストの意味するところである．

もしテクストを完全に電子化しようとするなら，そのテクストをデジタル化して，結果として，そのテクストについての私たちの一つかあるいは複数の解釈をエンコードするだろう．

これがTEIの核心である． このために，TEIはまだ私たちにとって重要な存在である，と私は考えている．

これまで見てきたように，TEIはやや珍しい歴史的瞬間の成り行きとして生まれた．文学研究者，言語学者，テクスト校訂に携わる人々，歴史家，アーキビスト，図書館司書，書誌学者，コンピュータ科学者など，普段はほとんど付き合いのない人々の間で興味と関心が一致したのである．なぜ，そのタイミングでこれらの異なる専門性を持った人々すべてが集結したのだろうか？ その理由の1つは，このタイミングが今日デジタルの転換（digital turn）として知られるものが始まった時期だったためである．この時期は，学術的な原資料が大規模にデジタル形式に変換された．これが，先に挙げた人々を集結させたのである．

デジタル以前の学問的世界がいかに分断されていたか，あるいは，諸分野間で分断された既存の学術界を反映するように，新たな言語の不一致，新たなバベルの塔が生まれることに私たちがいかに危機を感じていたか，今となっては思い出すことが難しい．ナンシーが言及した，まったく異質なコミュニティ同士が，新たな，デジタル共通語とも呼んでもいいような言語を創り出そうとするTEIの試みのために協働するに至ったことは，そのような言語の混乱を阻もうとする決意であった．これが，TEIがいろいろな場所でバベルの塔のイメージや比喩を用いる理由である．

ナンシーが分析してみせたように，学際性はTEIの取り組みにおける重要な特徴であった．作業部会のメンバーは，多様な専門分野を持っていた．正直に言えば彼らの多くは，デジタルの転換への関心と新たな言語の混乱を防ぐことへの切望を除いて，共通点はほとんどなかった．

そして，その多様性は，TEIのいくつかのより印象的なアーキテクチャの特徴を表している．そのうちのいくつかは，マイケルが既に述べたものである．私はそのうちの2点についてちょっと語ってみたい．それは，オッカムのカミソリの幅広い適用とカスタマイズの必要性への要求についてである．

オッカムは有名なオックスフォードの哲学者であり，「ある事柄を説明するためには，必要以上に多くを仮定するべきでない」と述べている．（彼が「必要」という言葉で指したものが何であったか説明があったならば，より明確だっただろう．）TEIはその問題に対処する必要があった．マイケルと私は，さまざまな学問分野の専門家の意見を聞く幸せなひととき（場合によっては一週間，一ヶ月間）を数多く過ごした．一方は，例えば赤チームと呼び，「赤みを表現するためのタグが必要である」と言う．他方では，青チームと呼び，「TEIは青みを表現するためのタグを提供することが不可欠です．」と言う．

マイケルと私は物の色を表現するためのタグを提案した．これで，物が赤であるか青であるかを表現することができる．しかし，もちろん，赤チームはそのようには考えず，青チームと同じタグを使いたがらなかった．なぜなら，結局のところ，赤と青は大きく異なり，赤チームと青チームは異なる学問分野に由来し，必ずしもそれほど交流しようとはしなかった．

しかし，可能な限り，TEIは色のような一般的な概念を定義する．それは，章，節，項，幕，場面，詩編，パートや，特定のジャンルまたは区切りに関することである．もしくはqのように TEI P5において，何らかの理由で周囲のテクストと引用符または同様の方法で区別される任意のマテリアルに使用されるものだ．これらは非常に高レベルなものであり，ほとんど何にでも適用できる．

しかし同時に，逆説的にも思われるが，ガイドラインは，一般的なエレメントの非常に特殊なバリエーションをしばしば含む．引用符の特定の種類の使用を示すsaid，quote，mentioned，soCalled要素がその例である．

私が議論したい第2の主要なアーキテクチャ上の特徴は，マイケルがすでに強調している，カスタマイズ性の重要さである． TEIはユーザにできあがった解決策を提供することはない．TEIはユーザにレゴキットのような，ユーザもしくはプロジェクトの特定のニーズに合わせた特定のエンコーディング方式を構築することができるフレームワークを提供している．そして，時間がたつにつれ，多くのユーザが独自のニーズに合わせて新しい種類のレゴのピースを追加し，それらの新しいピースの多くがガイドラインに統合される．

Text Encoding Initiativeという名前にも関わらず，TEIは，テクストだけに関心があるわけではない．それは先述したように，主に，そして本質的にテクストに関わるが，テクストだけに限定されるものではない． TEIの最初のバージョンであるTEI P1であっても，書誌メタデータ，様々な種類の抽象的な言語分析，書記体系の文書化のために提案されたマークアップが存在した．もちろん，従来の本の構造とその典型的な構成要素のためのマークアップも存在した．TEI P2では，書き起こされた音声やその他の特殊な情報の形態が追加された．

当初，TEIはWebのようなものについてはあまり言及しなかった（それが存在しなかったため）．それは，レイアウトや書式，きれいにプリントアウトするための方法についてはあまり言及していなかった．なぜなら，その仕事を完全に適切に実行する既存のシステムが存在し，それを再発明する必要がなかったからである．

デジタル画像とデジタル翻刻を組み合わせたデジタルファクシミリについては何も言及されていなかった．なぜなら，当時，50メガバイトの画像データとテクストを組み合わせるような余裕がなかったためである．また，人の名前やその他人への参照とは対照的に，人のような抽象的なオブジェクトを表現する方法については何も言及しなかった．なぜなら，それはデータベースの仕事だと感じていたためである．

そしてマイケルがすでに述べたように，ソフトウェアについては何も言及していなかった．

主に文学的および言語的パラダイムにおけるメタデータ，構造化されたテクスト，構造化されたテクストに対する分析に焦点を当てていた．

貧相に，見劣りするように見えるかもしれないが，実際にはそれが，TEIの発展と今日のものへと拡大するための非常に堅固な基礎を提供した． 今日のTEIモジュールは以下をカバーしている．

- テクストの構造的および機能的構成要素
- 歴史資料，画像の原本通りの翻刻と注釈
- 文書の構成要素間のあらゆる種類のリンク，対応，配置
- データおよび名前付きエンティティ：例えば，時間表現，人物，場所，およびイベントの名前，およびそれらが参照するエンティティ

- 本文以外のテクストへの注釈（訂正，抑制，追加）． 生成論，文書編集上の言語学的分析
- あらゆる種類のメタデータ
- マークアップ方式の正式な定義までも！

なぜあなたはまだTEIに注意を払う必要があるのだろうか？ なぜTEIはいまだ重要なのだろうか？なぜあなたはそれに知的アンテナを張っておくべきなのだろうか？

エジンバラ大学の計算言語学者ヘンリー・トンプソンは，標準化の提案が失敗する主な理由はおそらく2つあると示唆している．

まずそれらは，ある意味で未熟な時期かもしれない． それらは準備ができていないのだ．それらは早まった理論に基づいており，現実で正しくテストされていないか，または少数の人々とその友人にしか関心のない理論に基づいている． それが一つの理由である．

もう一つの理由は，標準化に関する提案が成熟した理論に基づいていたとしても，対象として意図されたユーザコミュニティはその提案に関心がなく，それを使用する意欲を感じないというものである． それは時に「NIH（Not Invented Here，ここで発明したものではない）症候群」と呼ばれる．

私としては，TEIの長寿の秘訣は，それがそれらの問題に対処したことだと考えている．

TEIがどのようにそれらに対処するかについて少し述べたい．ヘンリー・トンプソンが説明している2つの問題をTEIが回避している根本的な理由は，TEIが適応力が高いということだ．

使用される環境に対応して適応するTEIの能力は，おそらく，それを使って作業を続けてきた人々やそれを信じている人々，開発したいと思っている人々の動機や技能などの，より明白な要素と同じくらい重要である．

適応の重要な形態の1つは，TEI組織の意識的な非集中化である．TEIの最初の開発は，ナンシーが述べたように，3つのスポンサー組織が任命した中心的な運営委員会によって監督され，彼らは編集者を任命し，プロジェクト全体の管理を行った．編集者は，ガイドラインの全体的な学術的健全性の責任を担い，個々のワーキンググループの勧告を調整して全体をより調和させるために多くの時間を費やした． TEIが資するべきコミュニティに対する組織の責任感も強かったが，編集者と運営委員会にも，その責任（および付随する権限）が比較的強く集中していた．これが，研究プロジェクトを編成するよい方法であると考えている．ステークホルダーを特定し，その意見を参考にするが，物事を進める責任は，小規模な人々に属する．

現在のTEIコンソーシアムの組織では，責任感のある人が幾分増えており，TEIはコミュニティによってやや直接的に運営されている． TEI技術委員会は，ガイドラインをどのように編集，変更，維持するべきかについて決定を下し，TEI理事会は全体を組織する． そして，彼らはそれぞれコンソーシアムのメンバーによって選出される．また，理事会は本質的に，TEIコミュニティから直接の提案，またはspecial interest groupsとして編成されたコミュ

ニティの一部からの提案に応える． 現在，多くのオープンソースプロジェクトが同じようにしており，この組織に何も目立ったことはない．しかし，集中的に組織化された研究プロジェクトに適したガバナンスから，コミュニティによって維持されるインフラに適したガバナンスへの移行は， TEIの発展において生じた最も重要なことの1つであり，TEIが生き残り続けると私が考える理由の1つである．

TEIの適応性のもう一つの重要な部分は，もうすでに聞いていると思うが，カスタマイズの原則である． TEIのカスタマイズは，ほしい部分をつまみ食いすることだけではない．あなたが新しいものを追加することも，既存のものをさまざまな方法で変更することもできる．それはあなたが理論を成熟させる方法である． あなたはアイデアを成熟させるようにテストできる必要があり，TEIが提供するカスタマイズのための仕組みを使うことにより，それを容易に実現できる．

NIH症候群に関する限り，TEIは常に，世界についての複数の考え方を試し，吸収するために努力を傾けてきた． それはXML環境であるため，他のXML語彙と親和性が非常に高く，それらを簡単に統合することができる．例えば，ベクトルグラフィックスに対してSVGを，数式にMathMLを簡単に使用することができる．

ずいぶん前に，私は適切なTEIの使用の基本ルールを4つのルールで要約しようとした．

- 他のコード体系を持つことなかれ
  - これは，実際にはもう真実ではない． 他にもたくさんのコード体系がある．だから私たちはその戒めを破った．その代わりに私たちが現在持っている戒めは，「TEIを使用しているときと使用していないときを明確にする必要がある」ということである．
- 合意を讃えよ，汝がこの地で長く生きられるよう
  - これは今も正しいと思われる．
- 規格のGIをみだりに唱えるなかれ．
  - もちろん，GIは一般的な識別子，つまりエレメント名のことだ． これはまだ非常に重要な禁止事項である．
- 多義性を犯すなかれ．
  - 皆さんは大人の読者なので，多義性が何であるかは言及しない．

しかし，よく考えると，TEIに従うということは実際にはどういう意味なのだろうか？

もしTEIが常に拡大し続け，ますます複雑になり続け，もしあなたが好きなように変えることができ，あなたが好きなことを何でも行うことができ，もし何かを投げ捨て新しいものを入れることができれば， 一体何をもって「これはTEI文書です」と言えるのだろうか？

実際，それはかなり多くのことを意味する． つまり，文書内のタグ付け行為は，文書化されているものに従う．それは，あなたが考え方や概念，あるいはテクストについての理論の確立された語彙集を尊重していることを意味する．確立された合意事項を尊重し，それを使用している．あなたはテクストについて語るためにTEIが定義する言語を使っている．

しかし同時に，TEIに準拠したルールはあなたの独立性を尊重する． あなたはいつでも「TEIはこの種のマークアップを提供していますが，そのマークアップを使用するつもりはない． テクストの私の読みをよりよく反映する別のものを作り出すつもりだ． 」と言うことができる．

言い換えれば，TEIのために，標準化は「私があなたに言ったことをしなさい」，あるいは「私がやることをしなさい」ということを意味するものではない．標準化は，「われわれが理解できる言語を使って何をしたか教えてください．」ということを意味する．これは大きな違いである．

今日，私はTEI準拠の文書について，以下のように説明する．

- 整形式XML文書である．
- （やがて，XMLよりも優れた言語を思いつく人がいるかもしれない．その場合，TEIはこれを使用するようになるが，現時点では整形されたXML文書である）．
- スキーマに対しても妥当である．
- スキーマは，TEIによって定義されたすべての要素のサブセットでも，拡張機能でも構わないが，それは別の問いである． 文書がTEI名前空間の要素を使用する場合，ガイドラインに示されているように，それらの要素のセマンティクスを尊重する必要がある．
- そして，マイケルがすでに述べたように，TEIの使用法とTEIの拡張はすべて文書化されなければならず，TEIはマークアップ方式を文書化する目的で設計された特定のTEI語彙を提供する． それはあなたのシステムを文書化するために使用すべきものである．

これらのルールの重要性は，誰でもTEIを独自に修正し，文書化し，他のメンバーと協議し，ガイドラインの新機能を提案し，今後のガイドライン開発に役立てることができるということである． 新しいバージョンのTEI P5は，年に2回登場し，このように開発された作業を取り入れている．

もちろん，TEIガイドラインのメンテナンスを本当に手伝っていきたい場合は，個人としてコンソーシアムに参加したり，もしくは（もしあなたが機関に影響力のある人であれば）あなたの機関がコンソーシアムに参加するように説得したりできる．

では結論を述べよう．

TEIがまだ現役である本当の理由は何だろうか？

実際のところ，私の考えでは，まだそれが存続しているのはあなたがそう望むからである．

デジタル・ヒューマニティーズのコミュニティ全体が，私たちの話していたことをやめてしまえば，TEIは必要がなくなる． しかし，TEIが役立つようなことをあなたがやり続けたいと思う限りは，あなたはそれを使うことを望み続け，あなたの要求に合うように変更し続けるだろう．

言い換えれば，TEIがまだここにある理由は，それがコミュニティの賜物だからであろう．

この賞を受け取るように求められたのは3人だが，この賞の本当の受賞者はTEIコミュニティ全体である．TEI理事会の現在のメンバーは起立されたい．また，理事会，TEI作業部会またはSIGの過去のメンバー，TEIコンソーシアムの現在または過去のメンバー（個人または機関），およびTEIの現在または過去のユーザも起立されたい． そして最後に，「TEIは全く間違っている」と言うために，このような会議のセッションに立ち会ったことのある人は起立されたい．TEIが何か間違っていると思うほどマークアップを気にしている人もまた，TEIコミュニティのメンバーの一人である．

ありがとう．

テクスト・エンコーディングのガイドラインに向けて

ヴァッサー企画会議の締めくくりの言葉

文書番号: TEI PC P1

1987年11月13日，ニューヨーク，ポキプシー．

1. ガイドラインは，人文学研究におけるデータ交換のための標準的な形式を提供することを目指す．
2. ガイドラインは，同じ形式でテクストのデジタル化をするための原理を提案することも目指す．
3. ガイドラインは，以下のことをすべきである．
   - 形式に関して推奨される構文を定義する．
   - テクストデジタル化のスキーマの記述に関するメタ言語を定義する．
   - 散文とメタ言語の双方において新しい形式と既存の代表的なスキーマを表現する．
4. ガイドラインは，様々なアプリケーションに適したコーディングの規則を提案するべきである．
5. ガイドラインには，そのフォーマットにおいて新しいテクストを電子化するための最小限の規則が入っているべきである．
6. ガイドラインは，以下の小委員会によって起草され，主要なスポンサー組織の代表による運営委員会によってまとめられる．
   - テクスト記述
   - テクスト表現
   - テクスト解釈と分析
   - メタ言語定義と，既存・新規のスキーマの記述．
7. 既存の標準規格との互換性は可能な限り維持されるだろう．

8. 多くのテクスト・アーカイブズは，原則として，交換形式としてのそれらの機能に関して，そのガイドラインを支持することに賛成した．私たちは，この交換を効率化するためのツールの開発を援助するよう，支援組織に働きかける．
9. 既存の機械可読なテクストを新しい形式に変換することとは，それらの規則を新しい形式の構文に翻訳するということを意味しており，まだデジタル化されていない情報の追加に関して何か要求されるということはない．

［フランス語版］

1. このガイドラインは，人文学研究におけるデータ交換のための標準的な形式を提供することを目指す．
2. このガイドラインは，同じ形式でテクストのデジタル化をするための原則を提案することも目標とする．
3. ガイドラインは以下のことをすべきである．
   - 形式に関して推奨される構文を定義する．
   - テクストデジタル化のスキーマの記述に関するメタ言語を定義する．
   - 散文とメタ言語の双方において新しい形式と既存の代表的なスキーマを表現する．
4. ガイドラインは，様々なアプリケーションに適したコーディングの規則を提案するべきであろう．
5. ガイドラインには，そのフォーマットにおいて新しいテクストをデジタル化するための最小限の規則が入っているべきである．
6. ガイドラインの作成作業は，以下のテーマごとの小委員会に委任される．
   - テクスト記述
   - テクスト表現
   - テクスト解釈・分析
   - メタ言語定義と，既存・新規のスキーマの記述
7. これらの作業は，この取り組みを支える主要な団体の代表によって構成されるだろう運営委員会によってまとめられる．
8. 既存の標準規格との互換性は可能な限り維持される．
9. 大規模テクスト・アーカイブズの代表の多くが，原則的に，データ交換のための形式の記述として，そのガイドラインを使用することに賛成する．私たちは，この交換を効率化するためのツールの開発を援助するよう，支援組織に働きかける．
10. 既存の機械可読なテクストを新しい形式に変換することとは，それらの規則を新しい形式の構文に翻訳するということを意味しており，まだデジタル化されていない情報の追加に関して何か要求されるということはない．

（フランス語訳：P. A. フォティエ）

会議参加者

- Helen Aguera, National Endowment for the Humanities
- Robert A. Amsler, Bell Communications Research
- David T. Barnard, Queen's University, Kingston, Ontario
- Lou Burnard, Oxford University Computing Service
- Roy Byrd, IBM Research
- Nicoletta Calzolari, Istituto di Linguistica Computazionale C.N.R.
- David Chesnutt, University of South Carolina
- Yaacov Choueka, Bar-Ilan University, Ramat-Gan, Israel
- Jacques Dendien, Institut National de la Langue Française
- Paul A. Fortier, University of Manitoba, Winnipeg, Manitoba
- Thomas Hickey, Consulting Research Scientist, OCLC Online Computer Library Center
- Susan Hockey, Oxford University Computing Service
- Nancy M. Ide, Vassar College
- Stig Johansson, University of Oslo
- Randall Jones, Brigham Young University
- Robert Kraft, University of Pennsylvania
- Ian Lancashire, University of Toronto
- D. Terence Langendoen, City University of New York
- Charles (Jack) Meyers, National Endowment for the Humanities
- Junichi Nakamura, Kyoto University, Japan
- Wilhelm Ott, Universität Tübingen
- Eugenio Picchi, Istituto di Linguistica Computazionale C.N.R.
- Carol Risher, Association of American Publishers, Inc.
- Jane Rosenberg, National Endowment for the Humanities
- Jean Schumacher, CETEDOC
- J. Penny Small, Lexicon Iconographicum Mythologiae Classicae
- C. M. Sperberg-McQueen, University of Illinois at Chicago
- Paul Tombeur, Director, CETEDOC
- Frank Tompa (New OED Project, Waterloo), Bell Communications Research
- Donald E. Walker, Bell Communications Research
- Antonio Zampolli, Istituto di Linguistica Computazionale C.N.R.

[1] カンファレンス予稿集はLiterary data processing conference : proceedings of the conference organized by the Department of scientific and technical information, technology and engeneering staff, IBM Corporation, and held at the Thomas J. Watson Research Center, Yorktown Heights, New York, Sept. 9-11, 1964, ed. Jess B. Bessinge

r, Jr., and Stephen Maxfield Parrish (White Plains: IBM, 1965)として出版された．標準化に向けたケイの訴えは後に “Standards for encoding data in a natural language,” Computers and the Humanities 1.5 (1967): 170-77として刊行された．ケイの原稿はWeb上でスキャンを見ることができる．URLは以下：https://www.researchgate.net/publication/267793697_STANDARDS_FOR_ENCODING_LINGUISTIC_DATA.

[2] J. Coombs, A. H. Renear, and S. J. DeRose, “Markup systems and the future of scholarly text processing,” Communications of the Association for Computing Machinery 30.11 (1987): 933-947.

[3] IETFやW3Cの多くのワーキンググループでは，一つの仕様を開発するには，明晰さ，わかりやすさ，簡潔さというメリットをもたらす複数の実装がつきものである．しかし，理念的に複数の競合する実装が存在するゆえにこそ，それを実装するソフトウェアで開発されるIETFやW3Cの技術的仕様の混乱を避けやすくなっている．TEIの場合，複数の商用ソフトウェア・ベンダーが参画しているわけではなかったため，あらゆるソフトウェア開発はTEIによって行われなければならなかっただろうし，同じフォーマットで動作するソフトウェアは複数ではなく一つだけだっただろう．